An Advanced Diffie-Hellman Approach to Image Steganography

Shreyank N Gowda
Project Associate, Computer Science And Engineering Department,
IIT-M, India
kini5gowda@gmail.com

Abstract**—** Diffie-Hellman key exchange is one of the most popular key exchange algorithms used in cryptography. Steganography is the approach of hiding information in a medium. When that medium is an image it is called as image steganography. The most popular image steganography algorithm in use presently is the Least Significant bit Algorithm (LSB). However this algorithm has proved to be easy to crack and hence rather insecure. The proposed algorithm uses a variation of the Diffie-Hellman algorithm to determine the pixel positions of the image to embed the information in. Both the sender and receiver maintain private keys. Along with this a public key is shared between them. The sender sends a public key to the receiver and vice versa. This public key is nothing but the product of the private key with the public key. The shared secret key is then the public key received multiplied with the private key of each. Now this value obtained is considered to be the pixel to embed data in. So all pixels with multiples of this value are embedded first using LSB. If still some data persists, then the shared key is substracted by 1 and all multiples of this new value are then filled. The process is repeated till the entire information is hidden.

Keywords— Steganography;LSB;Diffie-Hellman

I. INTRODUCTION

Steganography is the process using which we can hide data of some type i.e images, videos ,text etc by making use of a medium as cover. The word steganography has been obtained from the Greek Language by making use of two words. The first word is "steganos" which translates to "being concealed or covered" and the second word "graphein" which translates to "writing". Steganography is often misunderstood with cryptography. Although they both serve the same purpose they both are different entities. Cryptography encrypts some information and delivers that without trying to hide the fact that something is being hidden. So one look at an encrypted text will bring the attention of the attacker as it becomes obvious that something is being hidden. Steganography meanwhile prevents attention being drawn to itself as an object of scrutiny since the attacker will not have a clue that there is the possibility of some information being hidden. Thus, whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned with concealing the fact that a secret message is being sent, as well as concealing the contents of the message.

The most basic algorithm used for steganography is the Least Significant Bit (LSB) algorithm. In the algorithm the least significant bit of each pixel from an image is modified so that it can hide one character of information from the plain text. This algorithm is not strong and can be easily broken down and any information being hidden using this algorithm can easily be obtained by an attacker. This paper uses a famous cryptography algorithm to solve the purpose of increasing the security of the information being hidden.

To understand LSB algorithm consider an example, For example a grid for 3 pixels of a 24-bit image can be as follows: 10111101 01101100 11010110 11100110 11010100 00011100 11111010 11101101 01101011 When the number 202, which binary representation is 11001010, is embedded into the least significant bits of this part of the image, the resulting grid is as follows: 10111101 01101101 11010110 11100110 11010101 00011101 11111010 11101101 01101010.

II. BACKGROUND

In [1] the author tells us the possibility of making use of a texture as a cover image in which we can possibly hide data. A texture is nothing but a set of pixels that go on repeating themselves. Hence it becomes much easier to use as a medium to hide information and we can hide a lot of information since we can go on repeating the texture. But a texture provides an indication that information is being hidden and hence is more vulnerable to attacks.

In [2] a mapping technique called the non-linear chaotic mapping technique has been used. The data that has to be hidden is first embedded on a scrambled image. Then the cover image is at the same time subject to Discrete Wavelet Transform. This new cover image is even further embedded along with the scrambled image. This method can be used to hide large quantity of data, however should there be a possibility of the map being obtained it is extremely feasible to obtain the data.

In [3] a method of hiding data utilising any of the RGB colour channels has been suggested. Any image has 3 channels for colours and hence changing value of either one of the channels

does not show much change visually. However this method cannot be used to hide large images.

In [4] two steps are taken, firstly Huffman encoding is done on the data and then the data is broken down to blocks. Simultaneously Discrete Cosine Transform is performed on the image being used as cover. LSB is then modified by using Huffman values obtained. Maintaining the Huffman codes are extremely important since loss of Huffman codes means loss of data. Also computation time for this method is relatively large.

In [5] double layer of security of the data is given, the first layer of security is done by using the standard Least Significant Bit method and the second layer involves the Advanced Encryption Standard Algorithm to encrypt the text before using it. Steganography does not replace the encryption of data, instead it provides extra security feature to it.

In [6] the message desired to be hidden will be made to embed in only one channel, the blue portion of the RGB channel. Results have verified that this enhanced the security capability of the image as visual distortion was not perceptible.

In [7] the secret data is taken and at first an encryption is done by using the Vigenere encryption method to increase the security level of the algorithm and also protection of the data. Next, the Lempel Ziv Welch (LZW) technique is used in compression of the data, this is done to hide its actual capacity. Afterwards, the extended knight tour algorithm is executed where each bit stream of data is made to spread out over the image. This increases the image's robustness.

In [8], a steganography algorithm based on Arnold Transform, discrete cosine transform and Chaotic System is proposed. The chaotic system ensures a random sequence is generated for spreading the data in the frequency band Discrete Cosine Transform coefficient of the cover image. The secret data is again further scrambled using the Arnold Cat Map which only enhances the security. The recovery process is done by doing the exact same method in reverse manner.

In [9] the idea of public key crypto-system was exhibited. The protocol proposed was the first system to use public key or two- key cryptography. In the protocol a secret key was used to exchange between two portions over a less secure channel without exchanging any prior information between them.

III. Proposed Algorithm

The proposed algorithm makes use of the idea behind the Diffie-Hellman algorithm to determine the pixel position to embed some information. Let us consider an example to understand the working of the algorithm before actually looking at the algorithm.

Let's assume there are two users A and B. Let's say that a private shared key is present around and it's value is 8. All keys do not have to be prime, as was the case of Diffie-Hellman. The fact that the numbers are not prime makes it even harder to factorise out the private keys for any intruder. Now consider A has a private key of 12. So A sends the value 96 (private multiplied with shared) to B over the insecure channel. Now B also has a private key, say 18. So B sends 144 to A. Now A gets the value of shared stego key as 144 multiplied with 12 which is 1728. B simultaneously calculates the key value as 96 multiplied with 18 which is the same, 1728. So now we can declare that 1728 is a shared stego key. We however assume that it is safe for both users to be knowing each other's private keys for a particular transaction. Also for every new transaction a new private key is generated and hence privacy is still maintained.

Next A takes the input image and the plaintext. It takes the first character of the plain text and embeds it onto the 1728^{th} pixel. The next character is hidden in the $1728x2^{th}$ pixel and so on. This process is repeated till all values ofr pixels having multiples of 1728 are filled. Once they are filled and if data to embed still remains, the 1727 pixel is taken and the process is repeated. This entire procedure repeats till all characters of data are embedded.

Now let's look at the algorithm,
Step 1: Start
Step 2: Choose private keys for A and B and a shared secret key for A and B
Step 3: Send value of private key multiplied with shared key from A to B and vice versa
Step 4: Multiply the obtained value with the private key to obtain a key called the shared stego key, let this value be called 'x'
Step 5: Now A takes input image and proceed to x^{th} pixel and embed character to LSB of that pixel
Step 6: Repeat for multiples of x
Step 7: If all multiples are covered and data still persists then set x = x - 1 and go to step 5 and repeat
Step 8: If all data has been embedded B can extract it out since it has the stego key
Step 9: Stop

The flowchart of the working of the process is shown in Fig. 1.

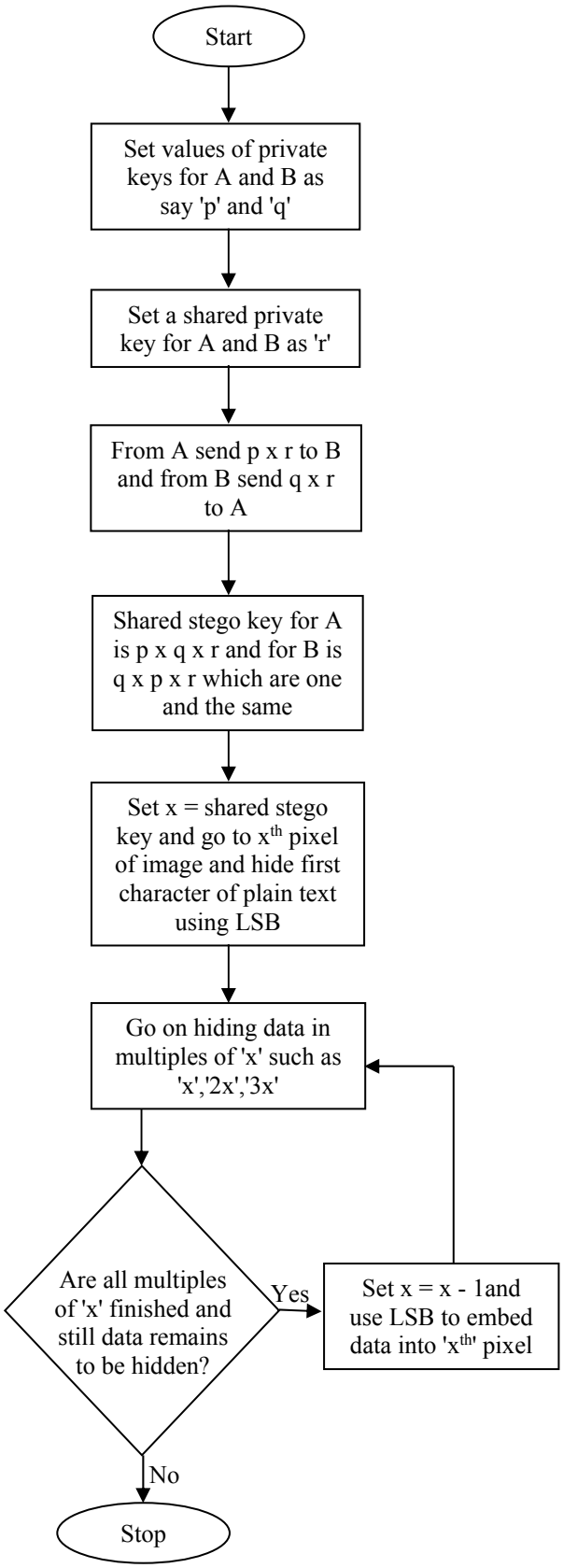

Figure 1. Flowchart of working

IV. Experimental Analysis

To determine the strength of the proposed algorithm. It was tested against the standard LSB algorithm using three testing parameters namely,

- maximum size of file that can be hidden
- time taken for execution
- PSNR value

Table I shows the results of maximum size of file that could be hidden in comparing both algorithms.

Table I. Result for determining maximum size that can be hidden

S.No	Input Size	Output for LSB	Output for proposed algorithm
1	320x240	512kB	512kB
2	640x480	2048kB	2048kB
3.	1280x780	8192kB	8192kB

As can be seen the maximum size of file that could be hidden are exactly the same. This is understandable because the number of pixels we use for embedding information is exactly the same in both cases.

Next Table II shows the results of time of execution of each algorithm for different sizes of files to be hidden. As can be seen from the table time taken to execute the standard LSB is lesser. As can be seen from results the time of execution is slightly lesser for the standard LSB. This is because no calculations has to be performed for LSB, whereas some calculation even though minimal is needed for the proposed algorithm.

.

Also it is very important to remember that in practical cases files of upto 100kB are hardly used for hiding these types of information. Hence this algorithm proves very useful.

Table II. Time of execution

S.No	Input Size (Image Size, Text size)	Output for LSB	Output for proposed algorithm
1	1280x780, 1kB	0.489 secs	0.511 secs
2	1280x780, 10kB	1.265 secs	1.298 secs
3	1280x780, 100kB	8.327 secs	8.411 secs

Table III presents the results of Peak Signal to Noise Ratio between the pre-algorithm image to the post-algorithm image. The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image

compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. This is used to indicate the maximum difference between the images. The Peak Signal to Noise Ratio is the difference between corresponding pixel values of the pre-algorithm to post-algorithm image. This is done using MatLab. The higher the value of PSNR the lesser is the difference in quality of the image.

Table III. PSNR value determination

S.No	Input Size (Image Size, Text size)	Output for LSB	Output for proposed algorithm
1	1280x780, 1kB	77.15	77.17
2	1280x780, 10kB	71.14	71.09
3	1280x780, 100kB	63.25	63.28

On a similar level the algorithm was tested for 32 more cases. The average of all the results were calculated and on average the PSNR value for proposed algorithm is 0.04 better than the standard LSB.

Fig. 2. shows a comparison of the image before it's execution by the algorithm and after. As can be seen visually there is no difference.

Figure 2 (a) Original Image (b) Image containing data

CONCLUSION

Based on the experimental analysis it is safe to say that in terms of comparison with the LSB algorithm the proposed algorithm gives exact results. However when it comes to security aspect of the algorithm the proposed algorithm gives much better results. The time needed for an attacker to first obtain the various keys involved in the proposed algorithm is a lot, because even factoring out the keys is hard due to them being non prime and also the fact that he needs 3 keys to start to extract information. Along with this the time needed to extract the information is relatively more due to the calculations involved. Hence it is safe to say that the proposed algorithm is much more efficient especially in terms of security of the information and also the algorithm.

ACKNOWLEDGMENT

Any work be it scholastic or otherwise does not depend solely on the person doing the work. Various people have given be intellectual, professional and emotional support during my time conducting this work. This is my opportunity to thank them all.

I would like to thank my family and my friends for being a strong emotional support throughout my time performing this project.

REFERENCES

[1] K. Wu and C. Wang, "*Steganography using reversible texture synthesis*" IEEE Transactions on Image Processing Vol.24 pp 130-139,January 2015

[2] S.Thenmozhi and M.Chandrasekaran, "*A novel technique for Image Steganography using Nonlinear Chaotic Map*",7th International Conference on Intelligent Systems and Control 2013 pp 307-311

[3] M. Parvez and A .Gutub, *"RGB Intensity Based Variablr-Bits Image Steganography"*,IEEE Asia-Pacific Services Computing Conference, pp 1322-1326,2008

[4] A.Nag, S.Biswas, D.Sarkar, P.P.Sarkar "*A novel technique for image steganography based on Block-DCT and Huffman Encoding*" International Journal Of Computer Science and Information Technology, pp 103-112, vol 2, June 2010

[5] S. Singh and V. K. Attri *Dual Layer Security of data using LSB Image Steganography Method and AES Encryption Algorithm* International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 8, No. 5 (2015), pp. 259-266

[6] S. Gupta , G. Gujral and N. Aggarwal *Enhanced Least Significant Bit algorithm For Image Steganography* IJCEM International Journal of Computational Engineering & Management, Vol. 15 Issue 4, July 2012 pp 40-42

[7] M. Bashardoost ,G. B. Sulongand, P. Gerami *Enhanced LSB Image Steganography Method By Using Knight Tour Algorithm, Vigenere Encryption and LZW Compression* IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013, pp. 221-227

[8] [S. Singh and T. J. Siddiqui *A Security Enhanced Robust Steganography Algorithm for Data Hiding* IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012 pp 131-139

[9] W. Diffie and M.E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 1976 , pp. 644-654.